

Introduction to Python I (Exercises 06)

Text Files + Exceptions

- 2) Write a small text file in notepad (maybe 3 lines)

Save the file somewhere in the pc where you have access. Attention sometimes the root of your c: drive is not accessible.

Open the file and display all the lines on the screen (one line at a time)

```
#if the file is in your current directory
myfile = open("textfile.txt",'r')
for line in myfile:
    print(line, end='')
myfile.close()
```

```
#you could also use the readline() method inside a loop.
filehandle = open("text01.txt", "r")
line = filehandle.readline()
while line != '':
    print(line, end='')
    line = filehandle.readline()

filehandle.close()
```

- 3) Write a program that asks the user for strings and writes them to a file. The program stops asking for strings if we type 'Q' or 'q'.

Check the file in Notepad or any other editor.

```
filehandle = open("text04.txt", "w")
while True:
    string_name = input('Enter a name: ("q" or "Q" to stop)')
    if string_name in "Qq":
        break

    filehandle.write(string_name + '\n')
filehandle.close()
```

- 4) Write a program that reads a text file and loads all the words in a dictionary. Keep count of the number of occurrences of each word in your dictionary. Assume words are delimited by spaces.

Print all the words in your dictionary with the number of times they occur.

```
myfile = open("somewords.txt",'r')
mycontent = myfile.read()

mydict = {}

#This next line automatically splits the contents of a string (space
#is use by default and it returns a list

wordlist = mycontent.split()

for word in wordlist:
    mydict[word] = mydict.get(word,0) + 1
```

cont ...

```
print()

print("Table of words and occurrences")
print("-----")

for key, value in sorted(mydict.items()):
    print("{:<10}: {}".format(key, value))

myfile.close()
```

- 1) Write a script that prompts for an integer. During the input you will convert the entered string into an integer (int function). Trap any exceptions and print a message letting the user know there was an error. Try entering letters instead of an integer, what happens? Are you able to trap the exception?

```
try:
    myinteger = int(input("Please enter an integer: "))
except:
    print("Invalid integer value ... !")
else:
    print(myinteger)
```

- 2) Try the same type of code but inside a function (call the function `inputint()`). The function will receive a string parameter containing the prompt to be displayed. When you call the function the function will return an integer inputted from the console. The function never exits until you enter an integer and it is successfully returned.

```
def inputint(prompt):
    integerFlag = True
    while integerFlag:
        try:
            myinteger = int(input(prompt))
        except:
            integerFlag = True
        else:
            integerFlag = False
    return myinteger

getinteger = inputint("Please enter an integer: ")
print(getinteger)
```

- 3) Write a simple script that divides a number (integer) by zero. Can you trap the exception and print a suitable message?

```
myinteger = 10
try:
    result = myinteger / 0 # This will cause an exception
except:
    print("Division by zero is not allowed!")
```

- 4) Prompt the user for a filename that does not exist. Try opening the file. Catch the exception and give a suitable error. Prompt for the file again or 'q' to exit.

```
filename = input("Enter file to be opened: ")
while filename != 'q':
    try:
        filehandle = open(filename, 'r')
    except:
        print("File {} does not exist!".format(filename))
    else:
        break
    filename = input("Enter file to be opened: ")
```

- 5) Write a script that prompts for an integer. During the input you will convert the entered string into an integer (int function). Trap any exceptions and print a message letting the user know there was an error. Try entering letters instead of an integer, what happens? Are you able to trap the exception?

```
try:
    myinteger = int(input("Please enter an integer: "))
except:
    print("Invalid integer value ... !")
else:
    print(myinteger)
```

- 6) Try the same type of code but inside a function (call the function inputint()). The function will receive a string parameter containing the prompt to be displayed. When you call the function the function will return an integer inputted from the console. The function never exits until you enter an integer and it is successfully returned.

```
def inputint(prompt):
    integerFlag = True
    while integerFlag:
        try:
            myinteger = int(input(prompt))
        except:
            integerFlag = True
        else:
            integerFlag = False
    return myinteger
getinteger = inputint("Please enter an integer: ")
print(getinteger)
```

- 7) Write a simple script that divides a number (integer) by zero. Can you trap the exception and print a suitable message?

```
myinteger = 10

try:
    result = myinteger / 0 # This will cause an exception
except:
    print("Division by zero is not allowed!")
```

- 8) Prompt the user for a filename that does not exist. Try opening the file. Catch the exception and give a suitable error. Prompt for the file again or 'q' to exit.

```
filename = input("Enter file to be opened: ")

while filename != 'q':
    try:
        filehandle = open(filename, 'r')
    except:
        print("File {} does not exist!".format(filename))
    else:
        break

    filename = input("Enter file to be opened: ")
```

```
import random

mylist = []

for i in range(25):
    mylist.append(random.randint(1,100))

# Calculating occurrence of each element dictionary to store occurrences
# This is just one of many ways of checking the duplicates. Note: there are
# better ways of doing this in python, but given the knowledge you have
# acquired so far, this will be a good way of obtaining the desired result.

occurdict = {}

for i in mylist:
    occurdict[i] = occurdict.get(i,0)+1

for key, value in occurdict.items():
    if value > 1:
        print(key)
```